

# **JOBSHEET VI**

## **SEARCHING**

Mata Kuliah : Algoritma dan Struktur Data

Dosen : **Mungki Astiningrum, S.T., M.Kom.**



**Ilham Dharma Atmaja**

**244107020220**

**Kelas : 1A**

**Absen : 14**

**PROGRAM STUDI TEKNIK INFORMATIKA**  
**JURUSAN TEKNOLOGI INFORMASI**  
**POLITEKNIK NEGERI MALANG TAHUN 2025**

## 6.1 PERCOBAAN

```
package Praktikum6;

import java.util.Scanner;

public class Mahasiswa14 {

    String nim;

    String nama;

    String kelas;

    double ipk;

    Mahasiswa14(String nim, String nama, String kelas, double ipk) {

        this.nim = nim;

        this.nama = nama;

        this.kelas = kelas;

        this.ipk = ipk;

    }

    void tampilInformasi() {

        System.out.println("NIM : " + nim);

        System.out.println("Nama : " + nama);

        System.out.println("Kelas : " + kelas);

        System.out.println("IPK : " + ipk);

    }

}

class MahasiswaBerprestasi14 {

    Mahasiswa14[] listMhs = new Mahasiswa14[5];

    int idx;

    void tambah(Mahasiswa14 m) {

        if (idx < listMhs.length) {

            listMhs[idx] = m;

            idx++;

        } else {

            System.out.println("Data sudah penuh");

        }

    }

}
```

```

    }

    void tampil() {
        for (Mahasiswa14 m : listMhs) {
            m.tampilInformasi();
            System.out.println("-----");
        }
    }

    int sequentialSearch(double cari) {
        int posisi = -1;
        for (int j = 0; j < listMhs.length; j++) {
            if (listMhs[j].ipk == cari) {
                posisi = j;
                break;
            }
        }
        return posisi;
    }

    void tampilkanPosisi(double ipk, int posisi) {
        if (posisi != -1) {
            System.out.println("Data mahasiswa dengan IPK: " + ipk + " ditemukan
pada indeks " + posisi);
        } else {
            System.out.println("Data " + ipk + " tidak ditemukan");
        }
    }

    void tampilDataSearch(double ipk, int posisi) {
        if (posisi != -1) {
            System.out.println("NIM\t: " + listMhs[posisi].nim);
            System.out.println("Nama\t: " + listMhs[posisi].nama);
            System.out.println("Kelas\t: " + listMhs[posisi].kelas);
            System.out.println("IPK\t: " + ipk);
        } else {
            System.out.println("Data " + ipk + " tidak ditemukan");
        }
    }

```

```

    }

}

}

class MahasiswaDemo14 {

    public static void main(String[] args) {

        int jumMhs = 5;

        MahasiswaBerprestasi14 list = new MahasiswaBerprestasi14();

        Scanner sc = new Scanner(System.in);

        for (int i = 0; i < jumMhs; i++) {

            System.out.println("Masukkan data mahasiswa ke-" + (i + 1));

            System.out.print("NIM : ");

            String nim = sc.nextLine();

            System.out.print("Nama : ");

            String nama = sc.nextLine();

            System.out.print("Kelas : ");

            String kelas = sc.nextLine();

            System.out.print("IPK : ");

            double ipk = sc.nextDouble();

            sc.nextLine();

            System.out.println("-----");

            list.tambah(new Mahasiswa14(nim, nama, kelas, ipk));

        }

        list.tampil();

        System.out.println("-----");

        System.out.print("Masukkan IPK mahasiswa yang dicari: ");

        double cari = sc.nextDouble();

        System.out.println("Menggunakan Sequential Search");

        int posisi = list.sequentialSearch(cari);

        list.tampilkanPosisi(cari, posisi);

        list.tampilDataSearch(cari, posisi);

    }

}

```

## 6.2 VERIFIKASI HASI PERCOBAAN

```
-----
Masukkan data mahasiswa ke-3
NIM : 333
Nama : tio
Kelas : 2
IPK : 3,6
-----
Masukkan data mahasiswa ke-4
NIM : 444
Nama : lia
Kelas : 2
IPK : 3,5
-----
Masukkan data mahasiswa ke-5
NIM : 555
Nama : fia
Kelas : 2
IPK : 3,3
-----
NIM : 111
Nama : adi
Kelas : 2
IPK : 3.6
-----
NIM : 222
Nama : ila
Kelas : 2
IPK : 3.0
-----
NIM : 333
Nama : tio
Kelas : 2
IPK : 3.6
-----
NIM : 444
Nama : lia
Kelas : 2
IPK : 3.5
-----
NIM : 555
Nama : fia
Kelas : 2
IPK : 3.3
-----
Masukkan IPK mahasiswa yang dicari: 3,5
Menggunakan Sequential Search
Data mahasiswa dengan IPK: 3.5 ditemukan pada indeks 3
NIM      : 444
Nama     : lia
Kelas   : 2
IPK      : 3.5
PS C:\Users\TILHAM_DHARMA_A\OneDrive\Documents\SEMESTER 2 GACOR\Algoritma dan St
```

## 6.3 JAWABAN PERTANYAAN

1. Perbedaan antara `tampilDataSearch` dan `tampilPosisi` Pada class `MahasiswaBerprestasi` yaitu:
  - `tampilDataSearch`  
Tugasnya menampilkan isi data mahasiswa yang ditemukan, misalnya NIM, nama, Ipk, dan kelas
  - `tampilPosisi`  
Tugasnya hanya menampilkan posisi Dimana data yang dicari ditemukan dalam array
2. Fungsi `break` dalam potongan kode yaitu:
  - `Break` menghentikan loop paling dekat secara seketika
  - `Break` memastikan program langsung keluar dari `for/while` loop, sehingga efisiensi pencarian meningkat

## 6.2.1 Percobaan Binary Search

```
package Praktikum6;

import java.util.Scanner;

public class Mahasiswa14 {

    String nim;

    String nama;

    String kelas;

    double ipk;

    Mahasiswa14(String nim, String nama, String kelas, double ipk) {

        this.nim = nim;

        this.nama = nama;

        this.kelas = kelas;

        this.ipk = ipk;

    }

    void tampilInformasi() {

        System.out.println("NIM : " + nim);

        System.out.println("Nama : " + nama);

        System.out.println("Kelas : " + kelas);

        System.out.println("IPK : " + ipk);

    }

}

class MahasiswaBerprestasi14 {

    Mahasiswa14[] listMhs = new Mahasiswa14[5];

    int idx;

    void tambah(Mahasiswa14 m) {

        if (idx < listMhs.length) {

            listMhs[idx] = m;

            idx++;

        } else {

            System.out.println("Data sudah penuh");

        }

    }

}
```

```
}
```

```
void tampil() {  
    for (Mahasiswa14 m : listMhs) {  
        m.tampilInformasi();  
        System.out.println("-----");  
    }  
}
```

```
int sequentialSearch(double cari) {  
    int posisi = -1;  
    for (int j = 0; j < listMhs.length; j++) {  
        if (listMhs[j].ipk == cari) {  
            posisi = j;  
            break;  
        }  
    }  
    return posisi;  
}
```

```
int findBinarySearch(double cari, int left, int right) {  
    if (right >= left) {  
        int mid = left + (right - left) / 2;  
        if (listMhs[mid].ipk == cari) {  
            return mid;  
        }  
        if (listMhs[mid].ipk > cari) {  
            return findBinarySearch(cari, left, mid - 1);  
        }  
        return findBinarySearch(cari, mid + 1, right);  
    }  
    return -1;  
}
```

```
void tampilkanPosisi(double ipk, int posisi) {  
    if (posisi != -1) {
```

```
        System.out.println("Data mahasiswa dengan IPK: " + ipk + " ditemukan  
pada indeks " + posisi);
```

```
    } else {  
        System.out.println("Data " + ipk + " tidak ditemukan");  
    }  
}
```

```
void tampilDataSearch(double ipk, int posisi) {  
    if (posisi != -1) {  
        System.out.println("NIM\t: " + listMhs[posisi].nim);  
        System.out.println("Nama\t: " + listMhs[posisi].nama);  
        System.out.println("Kelas\t: " + listMhs[posisi].kelas);  
        System.out.println("IPK\t: " + ipk);  
    } else {  
        System.out.println("Data " + ipk + " tidak ditemukan");  
    }  
}  
}
```

```
class MahasiswaDemo14 {  
    public static void main(String[] args) {  
        int jumMhs = 5;  
        MahasiswaBerprestasi14 list = new MahasiswaBerprestasi14();  
        Scanner sc = new Scanner(System.in);  
  
        for (int i = 0; i < jumMhs; i++) {  
            System.out.println("Masukkan data mahasiswa ke-" + (i + 1));  
            System.out.print("NIM : ");  
            String nim = sc.nextLine();  
            System.out.print("Nama : ");  
            String nama = sc.nextLine();  
            System.out.print("Kelas : ");  
            String kelas = sc.nextLine();  
            System.out.print("IPK : ");  
            double ipk = sc.nextDouble();  
            sc.nextLine();  
            System.out.println("-----");  
        }  
    }  
}
```



```

        list.tambah(new Mahasiswa14(nim, nama, kelas, ipk));
    }

    list.tampil();
    System.out.println("-----");

    System.out.print("Masukkan IPK mahasiswa yang dicari (Sequential Search):
");
    double cari = sc.nextDouble();
    System.out.println("Menggunakan Sequential Search");
    int posisi = list.sequentialSearch(cari);
    list.tampilkanPosisi(cari, posisi);
    list.tampilDataSearch(cari, posisi);

    System.out.println("-----");
    System.out.println("Pastikan data sudah terurut untuk Binary Search!");
    System.out.print("Masukkan IPK mahasiswa yang dicari (Binary Search): ");
    cari = sc.nextDouble();
    System.out.println("Menggunakan Binary Search");
    int posisiBinary = list.findBinarySearch(cari, 0, jumMhs - 1);
    list.tampilkanPosisi(cari, posisiBinary);
    list.tampilDataSearch(cari, posisiBinary);
}
}

```

## 6.2.2 VERIFIKASI HASIL PERCOBAAN

```
-----
NIM : 333
Nama : lia
Kelas : 2
IPK : 3.3
-----
NIM : 333
Nama : lia
Kelas : 2
IPK : 3.3
-----
Kelas : 2
IPK : 3.3
-----
IPK : 3.3
-----
NIM : 444
Nama : susi
-----
NIM : 444
Nama : susi
Kelas : 2
IPK : 3.5
Nama : susi
Kelas : 2
IPK : 3.5
Kelas : 2
IPK : 3.5
-----
NIM : 555
IPK : 3.5
-----
NIM : 555
-----
NIM : 555
Nama : anita
Kelas : 2
NIM : 555
Nama : anita
Kelas : 2
Nama : anita
Kelas : 2
IPK : 3.7
-----
Kelas : 2
IPK : 3.7
-----
IPK : 3.7
-----
-----
Masukkan IPK mahasiswa yang dicari (Sequential Search): 3,7
Menggunakan Sequential Search
Data mahasiswa dengan IPK: 3.7 ditemukan pada indeks 4
NIM      : 555
Nama     : anita
Kelas   : 2
IPK      : 3.7
-----
Pastikan data sudah terurut untuk Binary Search!
Masukkan IPK mahasiswa yang dicari (Binary Search): []
```

## 6.2.3 PERTANYAAN

1. `int mid = left + (right - left) / 2;`
2. `if (listMhs[mid].ipk > cari) {`  
`return findBinarySearch(cari, left, mid - 1); // Pencarian di bagian`  
`kiri`  
`}`  
`return findBinarySearch(cari, mid + 1, right); // Pencarian di bagian kanan`
3. Program masih berjalan, tetapi hasil dari binary search tidak bisa akurat

#### 4. Hasil binary tidak sesuai maka algoritma harus dimodifikasi

```
int findBinarySearch(double cari, int left, int right) {
    if (right >= left) {
        int mid = left + (right - left) / 2;
        if (listMhs[mid].ipk == cari) {
            return mid;
        }
        if (listMhs[mid].ipk < cari) {
            return findBinarySearch(cari, left, mid - 1);
        }
        return findBinarySearch(cari, mid + 1, right);
    }
    return -1;
}
```

#### 5. Modifikasi di main

```
System.out.print("Masukkan jumlah mahasiswa: ");
int jumMhs = sc.nextInt();
sc.nextLine(); // Consume newline
MahasiswaBerprestasi14 list = new MahasiswaBerprestasi14(jumMhs);
```

Modifikasi di class MahasiswaBerprestasi14

```
class MahasiswaBerprestasi14 {
    Mahasiswa14[] listMhs;
    int idx;

    MahasiswaBerprestasi14(int jumlahMahasiswa) {
        listMhs = new Mahasiswa14[jumlahMahasiswa];
        idx = 0;
    }
}
```

### 6.3.1 PRAKTIKUM

```
package Praktikum6;

import java.util.Scanner;

class Dosen14 {
    String kode;
    String nama;
    String jenisKelamin;
    int usia;

    Dosen14(String kode, String nama, String jenisKelamin, int usia) {
        this.kode = kode;
        this.nama = nama;
        this.jenisKelamin = jenisKelamin;
        this.usia = usia;
    }

    void tampil() {
        System.out.println("Kode: " + kode);
        System.out.println("Nama: " + nama);
        System.out.println("Jenis Kelamin: " + jenisKelamin);
        System.out.println("Usia: " + usia);
    }
}

class DataDosen {
    Dosen14[] dataDosen = new Dosen14[10];
    int idx = 0;

    void tambah(Dosen14 dsn) {
        if (idx < dataDosen.length) {
            dataDosen[idx] = dsn;
            idx++;
        } else {
            System.out.println("Data sudah penuh");
        }
    }
}
```

```

    }
}

void tampil() {
    for (int i = 0; i < idx; i++) {
        dataDosen[i].tampil();
        System.out.println("-----");
    }
}

void sortingASC() {
    for (int i = 0; i < idx - 1; i++) {
        for (int j = 1; j < idx - i; j++) {
            if (dataDosen[j].usia < dataDosen[j - 1].usia) {
                Dosen14 temp = dataDosen[j];
                dataDosen[j] = dataDosen[j - 1];
                dataDosen[j - 1] = temp;
            }
        }
    }
}

void sortingDSC() {
    for (int i = 0; i < idx - 1; i++) {
        int idxMax = i;
        for (int j = i + 1; j < idx; j++) {
            if (dataDosen[j].usia > dataDosen[idxMax].usia) {
                idxMax = j;
            }
        }
        Dosen14 temp = dataDosen[idxMax];
        dataDosen[idxMax] = dataDosen[i];
        dataDosen[i] = temp;
    }
}

void insertionSortDSC() {

```

```

        for (int i = 1; i < idx; i++) {
            Dosen14 temp = dataDosen[i];
            int j = i;
            while (j > 0 && dataDosen[j - 1].usia < temp.usia) {
                dataDosen[j] = dataDosen[j - 1];
                j--;
            }
            dataDosen[j] = temp;
        }
    }
}

public class Main {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        DataDosen dataDosen = new DataDosen();
        int pilihan;

        do {
            System.out.println("Menu:");
            System.out.println("1. Tambah data dosen");
            System.out.println("2. Tampil data dosen");
            System.out.println("3. Sorting ASC (usia termuda ke tertua)");
            System.out.println("4. Sorting DSC (usia tertua ke termuda)");
            System.out.println("5. Keluar");
            System.out.print("Pilih menu: ");
            pilihan = scanner.nextInt();
            scanner.nextLine();

            switch (pilihan) {
                case 1:
                    System.out.print("Kode: ");
                    String kode = scanner.nextLine();
                    System.out.print("Nama: ");
                    String nama = scanner.nextLine();
                    System.out.print("Jenis Kelamin: ");
                    String jenisKelamin = scanner.nextLine();

```

```

        System.out.print("Usia: ");
        int usia = scanner.nextInt();
        scanner.nextLine();

        Dosen14 dsn = new Dosen14(kode, nama, jenisKelamin, usia);
        dataDosen.tambah(dsn);
        break;
case 2:
    dataDosen.tampil();
    break;
case 3:
    dataDosen.sortingASC();
    System.out.println("Data dosen setelah diurutkan (ASC):");
    dataDosen.tampil();
    break;
case 4:
    System.out.println("Pilih metode sorting:");
    System.out.println("1. Selection Sort");
    System.out.println("2. Insertion Sort");
    int metode = scanner.nextInt();
    scanner.nextLine();

    if (metode == 1) {
        dataDosen.sortingDSC();
    } else if (metode == 2) {
        dataDosen.insertionSortDSC();
    } else {
        System.out.println("Metode tidak valid");
    }

    System.out.println("Data dosen setelah diurutkan (DSC):");
    dataDosen.tampil();
    break;
case 5:
    System.out.println("Keluar dari program.");
    break;
default:

```

```
        System.out.println("Pilihan tidak valid.");  
    }  
    } while (pilihan != 5);  
  
    }  
}
```

### **6.3.2 LINK GITHUB**

**<https://github.com/ianmen10/SEMESTER-genap2.git>**